

Blatt 6 - Gruppe: G1-07

Mike Lenz, Jonas Tesfamariam

11. Juni 2023

Aufgabe 1

```
SELECT DISTINCT
    s.name,
    s.semester,
    avg(p.note) OVER (PARTITION BY s.semester) AS avgnote
FROM
    studenten s
JOIN pruefen p ON s.matrnr = p.matrnr
WHERE
    s.semester > 1;

WITH profUnbestanden (
    name,
    anzahl
) AS (
    SELECT DISTINCT
        p1.name,
        count(*) OVER (PARTITION BY p1.name) AS anzahl
    FROM
        professoren p1
    JOIN pruefen pr ON p1.persnr = pr.persnr
    WHERE
        pr.note = 5.0
) SELECT DISTINCT
    p.name,
    sum(pu.anzahl) OVER (ORDER BY pu.anzahl DESC) AS unbestanden
FROM
    professoren p
```

```
      JOIN profUnbestanden pu ON p.name = pu.name
ORDER BY
      unbestanden;
```

Aufgabe 2

```
WITH semesterAvg (  
    semester,  
    averageNote  
) AS (  
    SELECT DISTINCT  
        s1.semester,  
        avg(p.note)  
    FROM  
        studenten s1  
        JOIN pruefen p ON s1.matrnr = p.matrnr  
    GROUP BY  
        s1.semester  
) SELECT DISTINCT  
    s.name,  
    s.semester,  
    sa.averageNote  
FROM  
    studenten s,  
    semesterAvg sa  
WHERE  
    s.semester = sa.semester;  
  
WITH profUnbestanden (  
    name,  
    anzahl  
) AS (  
    SELECT DISTINCT  
        p1.name,  
        count(*) AS anzahl  
    FROM  
        professoren p1  
        JOIN pruefen pr ON p1.persnr = pr.persnr  
    WHERE  
        pr.note = 5.0  
    GROUP BY  
        p1.name  
)  
SELECT
```

```
p.name,  
(  
    SELECT  
        sum(pu1.anzahl)  
    FROM  
        profUnbestanden pu1  
    WHERE  
        pu.anzahl <= pu1.anzahl  
    ) AS unbestanden  
FROM  
    professoren p,  
    profUnbestanden pu  
WHERE  
    p.name = pu.name  
ORDER BY  
    unbestanden;
```

Aufgabe 3

```
CREATE MATERIALIZED VIEW notenpartner (  
    p1,  
    p2)  
AS (  
    SELECT DISTINCT  
        s1.matrnr AS student1,  
        s2.matrnr AS student2  
    FROM  
        studenten s1  
    JOIN pruefen p1 ON s1.matrnr = p1.matrnr,  
        studenten s2  
    JOIN pruefen p2 ON s2.matrnr = p2.matrnr  
    WHERE  
        NOT p1.matrnr = p2.matrnr  
        AND s1.semester = s2.semester  
        AND p1.persnr = p2.persnr  
        AND p1.vorlnr = p2.vorlnr  
        AND p1.note - p2.note <= 0.3  
        AND p1.note - p2.note >= - 0.3  
    )  
WITH RECURSIVE transitivPartner (  
    matrnr,  
    iter  
    ) AS (  
        SELECT DISTINCT  
            p2,  
            1  
        FROM  
            notenpartner  
        WHERE  
            p1 = 94823  
        UNION ALL SELECT DISTINCT  
            np.p2,  
            tp.iter + 1  
        FROM  
            notenpartner np,  
            transitivPartner tp  
        WHERE
```

```
        np.p1 = tp.matnr
        AND NOT np.p2 = tp.matnr
        AND tp.iter < 10
    ) SELECT DISTINCT
        s.name
FROM
    transitivPartner tp
    JOIN studenten s ON tp.matnr = s.matnr;

WITH RECURSIVE transitivPartner (
    matnr,
    iter
) AS (
    SELECT DISTINCT
        p2,
        1
    FROM
        notenpartner
    WHERE
        p1 = 94823
    UNION ALL SELECT DISTINCT
        np.p2,
        tp.iter + 1
    FROM
        notenpartner np,
        transitivPartner tp
    WHERE
        np.p1 = tp.matnr
        AND NOT np.p2 = tp.matnr
        AND tp.iter < 10
    ) SELECT DISTINCT
        count(DISTINCT s.name)
FROM
    transitivPartner tp
    JOIN studenten s ON tp.matnr = s.matnr;
```

Aufgabe 4

a)

i)

Das Tupel kann in v1 und v2 eingefügt werden, da die Bedingungen $b \leq 20$ sowie $a \leq 10$ in der View Definition erfüllt sind. Das Tupel kann jedoch nicht in v3 eingefügt werden, da die Bedingung $b \leq 5$ verletzt wird.

ii)

Das Tupel kann wieder in v1 eingefügt werden, da die Bedingung in der View Definition erfüllt ist. Das Tupel kann auch in v2 eingefügt werden, jedoch wird hier nicht geprüft ob $b \leq 20$ ist. Das Tupel kann nicht in v3 eingefügt werden, da die Bedingung $b \leq 5$ wieder verletzt wird.

iii)

Das Tupel kann in v1 eingefügt werden. Das Tupel kann nicht in v2 eingefügt werden, da die Bedingung $a \leq 10$ verletzt wird. Das Tupel kann in v3 eingefügt werden, da die Bedingung $b \leq 5$ erfüllt ist. Die kaskadierende check option macht hier keinen Unterschied.

iv)

Das Tupel kann nicht in v1 eingefügt werden, da die Bedingung $b \leq 20$ verletzt wird. Das Tupel kann in v2 eingefügt werden, da die Bedingung $a \leq 10$ erfüllt ist. Das Tupel kann nicht in v3 eingefügt werden, da die Bedingung $b \leq 5$ verletzt wird. Die lokale check option macht hier auch keinen Unterschied.

b)

```
CREATE TABLE pruefen (  
    matrnr INT,  
    vorlnr INT,  
    persnr INT,  
    note NUMERIC(2,1) CHECK (note IN (1.0, 1.3, 1.7, 2.0, 2.3, 2.7,  
                                     3.0, 3.3, 3.7, 4.0, 5.0)),  
    FOREIGN KEY (matrnr) REFERENCES studenten (matrnr),  
    FOREIGN KEY (vorlnr) REFERENCES vorlesungen (vorlnr),  
    FOREIGN KEY (persnr) REFERENCES professoren (persnr),  
);
```